Implementation of a Multilevel Wiki for Cross-Domain Collaboration

Kar Leong Ong, Thuy Nguyen and Cynthia Irvine
Naval Postgraduate School, Monterey, CA, USA
karleong_ong@yahoo.com.sg
tdnguyen@nps.edu
irvine@nps.edu

Abstract: The pace of modern warfare requires tools that support intensive, ongoing collaboration between participants. Wiki technology provides a hypertext content-based collaborative authoring and information sharing environment that includes the ability to create links to other web contents, relative stability, ease of use, and logging features for tracking contributions and modifications. Military environments impose a requirement to enforce national policies regarding authorized access to classified information while satisfying the intent of wikis to provide an open context for content sharing. The Global Information Grid (GIG) vision calls for a highly flexible multilevel environment. The Monterey Security Architecture (MYSEA) Test-bed provides a distributed high assurance multilevel networking environment where authenticated users securely access data and services at different classification levels. The MYSEA approach is to provide users with unmodified commercial-off-the-shelf office productivity tools while enforcing a multilevel security (MLS) policy with high assurance. The extensible Test-bed architecture is designed with strategically placed trusted components that comprise the distributed TCB, while untrusted commercial clients support the user interface.

We have extended the collaboration capabilities of MYSEA through the creation of a multilevel wiki. This wiki permits users who access the system at a particular sensitivity level to read and post information to the wiki at that level. Users at higher sensitivity levels may read wiki content at lower security levels and may post information at the higher security level. The underlying MLS policy enforcement mechanisms prevent low users from accessing higher sensitivity information. The multilevel wiki was created by porting a publicly available wiki engine to run on the high assurance system hosting the MYSEA server. A systematic process was used to select a wiki for the MYSEA environment. TWiki was chosen. To simplify identification of errors that might arise in the porting process, a three-stage porting methodology was used. Functional and security tests were performed to ensure that the wiki engine operates properly while being constrained by the underlying policy enforcement mechanisms of the server. An objective in designing the test plans was to ensure adequate test coverage, while avoiding a combinatoric explosion of test cases. Repeatable regression testing procedures were also produced. A conflict between the application-level DAC policy of the wiki and that of the MYSEA server was identified and resolved.

Keywords: Wiki, multilevel security, access controls, porting methodology

1. Introduction

The tempo of modern military operations is such that tools for computer-supported cooperative work (Carstensen 1999) are needed. Web-based collaboration is attractive because users merely require a browser to access evolving documents hosted on a server. Since their introduction in 1994, wikis represent a mass collaboration tool that allows users to view, add, edit and delete the content of a website (Wikipedia 2007a). Yet, in a multilevel security (MLS) environment where document classifications and user clearances constrain access to information, extra care must be taken to ensure that wiki solutions are usable while still conforming to security policy. We describe a design of an MLS-enabled wiki. Multilevel technology permits collaborators with different security attributes in a coalition environment to maximize information sharing while still adhering to the constraints of the overall security policy. Multilevel-aware instances of the wiki execute as untrusted subjects within the context of a multilevel architecture. High assurance policy enforcement ensures that wiki users logged in at high sensitivity levels are able to read and post information at their level, but are only able to read information at lower sensitivity levels. Correspondingly, users at lower sensitivity levels are only allowed access to less sensitive information. Before exploring the challenges of constructing a MLS wiki, a brief review of wikis and the target MLS environment is appropriate.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Implementation of a Multilevel Wiki for Cross-Domain Collaboration	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School ,Center for Information Systems Security Studies and Research (NPS CISR),Department of Computer Science,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
3rd International Conference on Information Warfare and Security (ICIW 2008), April 2008, Omaha, Nebraska, USA, pp. 293-304

14. ABSTRACT

The pace of modern warfare requires tools that support intensive, ongoing collaboration between participants. Wiki technology provides a hypertext content-based collaborative authoring and information sharing environment that includes the ability to create links to other web contents, relative stability, ease of use, and logging features for tracking contributions and modifications. Military environments impose a requirement to enforce national policies regarding authorized access to classified information while satisfying the intent of wikis to provide an open context for content sharing. The Global Information Grid (GIG) vision calls for a highly flexible multilevel environment. The Monterey Security Architecture (MYSEA) Test-bed provides a distributed high assurance multilevel networking environment where authenticated users securely access data and services at different classification levels. The MYSEA approach is to provide users with unmodified commercial-off-the-shelf office productivity tools while enforcing a multilevel security (MLS) policy with high assurance. The extensible Test-bed architecture is designed with strategically placed trusted components that comprise the distributed TCB, while untrusted commercial clients support the user interface. We have extended the collaboration capabilities of MYSEA through the creation of a multilevel wiki. This wiki permits users who access the system at a particular sensitivity level to read and post information to the wiki at that level. Users at higher sensitivity levels may read wiki content at lower security levels and may post information at the higher security level. The underlying MLS policy enforcement mechanisms prevent low users from accessing higher sensitivity information. The multilevel wiki was created by porting a publicly available wiki engine to run on the high assurance system hosting the MYSEA server. A systematic process was used to select a wiki for the MYSEA environment. TWiki was chosen. To simplify identification of errors that might arise in the porting process, a three-stage porting methodology was used. Functional and security tests were performed to ensure that the wiki engine operates properly while being constrained by the underlying policy enforcement mechanisms of the server. An objective in designing the test plans was to ensure adequate test coverage, while avoiding a combinatoric explosion of test cases. Repeatable regression testing procedures were also produced. A conflict between the application-level DAC policy of the wiki and that of the MYSEA server was identified and resolved.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**12**	

1.1 Wikis

Most commonly implemented as server-side scripting technology (Arronsson 2007), a wiki engine manages a set of documents known as wiki pages. These are written in plain text and stored in either a regular file system or a database. When a browser requests a page, wiki scripts translate the wiki page into HTML and returns it to the browser. Most wiki engines use a simplified, non-standardized markup language indicate various structural and visual conventions for the wiki pages. Written in HTML, each wiki page contains a header with links to scripts that support specific functionality. For example, invocation of the "edit" link presents the wiki page enclosed in a text-area form. Users edit the text and submit new versions to the website via an accompanying "save" link. The wiki engine usually supports versioning through a revision control system, and maintains a "history" log of recent changes.

1.2 MYSEA

The Monterey Security Architecture (MYSEA) is intended to enable the aggregation of data and services at different security classification levels into a distributed multilevel secure network. It is accomplished through the use of commercial-off-the-shelf (COTS) products together with judicious use of secure high assurance components that enforce a multilevel security (MLS) policy. The high assurance products support logical separation and controlled sharing across various sensitivity levels without relying on physical separation (Nguyen et al. 2005, Irvine et al. 2004, Bradney 2006). The MYSEA MLS server supports HTTP, IMAP and SMTP protocols. HTTP support allows a wide range of applications such as the Tarantella web enablement software and WebDAV to be used. WebDAV supports remote access and modification of files in the MLS context, whereas a wiki, enables rapid and distributed authoring of shared content. The Linux-like interface of the BAE System XTS-400 STOP operating system (Getronix 2002) allows many existing Linux applications to be ported without significant modification. The XTS-400 server enforces both discretionary access control (DAC) and mandatory access control (MAC) security policies. A relevant XTS-400 feature is its implementation of deflection directories.

Because many applications use standard directories such as */tmp* during processing, a way to permit applications to access such standard directories without causing overt and covert information leakage (Lampson 1973) is necessary. As a solution, deflection directories are instantiated at different security levels simultaneously and the OS automatically selects or creates an instance of the directory corresponding to the user's current session level. No data is shared between different deflection directory instances, and it is impossible for a process at a particular level to access instances of the directory at other levels. With this background, an analysis and application of criteria for selecting from among the many wikis now available the one best suited to meet the requirements of our multilevel environment is presented. A concept of operations is used to clarify the intended wiki functionality, which the resulting architecture is designed to support. Carefully designed and executed tests demonstrated that the implementation meets is objectives. Finally, we present a brief summary and discussion of related and future work.

2. Wiki engine selection

A major step in the implementation of a wiki for the multilevel environment was to choose an appropriate wiki engine.Approximately 140 different wiki engines, implemented using a wide range of programming languages, each with differing capabilities are available (Cunningham 2007). We chose to build on existing comparisons of wikis (Wikimatrix 2007, Cunningham 2007, Wikipedia 2007c) to identify initial selection criteria. Those features of particular interest in the context of our multilevel architecture were heavily weighted.

The paramount criterion was the ability of the wiki to be integrated into the MYSEA environment. Specifically, it must run on the RedHat 8.0 operating system and must be able to integrate with Apache HTTP Server Version 1.3.34. Any wiki that relied upon special purpose databases was also eliminated. Maintainability was another key consideration. The Top-10 wiki engine list (Cunningham 2007) was used as a proxy measurement for wiki engine maintainability. The rationale was that more popular engines would enjoy more usage and hence a greater likelihood of enhancement and support.

Support for essential wiki engine functionality was also a factor. For comparative analysis, the popular MediaWiki engine was used as the baseline against which the functionality of each candidate was compared. Using these criteria, the list of candidates was narrowed to two: PmWiki and TWiki. PmWiki is a server-side wiki engine with an implementation based on PHP (PmWiki 2007). TWiki, also a server-side wiki engine, is implemented using Perl (TWiki 2007b). Table 1 presents the detailed selection decisions and the elimination process. Eliminated wikis are shown with shaded entries in one or more columns.

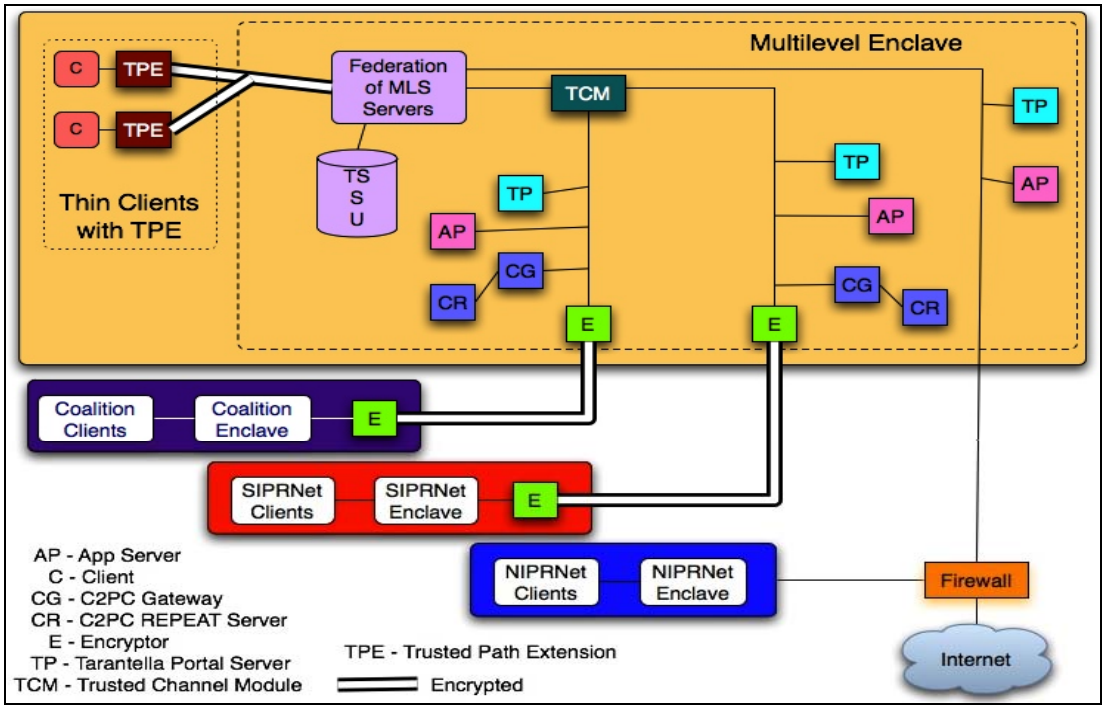

Figure 1: MYSEA test-bed topology [From (Nguyen 2005)]

Table 1: Selection decisions and elimination process

		Elimination Process							
	Wiki Engines	Regular File System (1)	Top 10 Wiki List (2)	Interface with Apache (3)	Essential Functionality (3)	ACL (3)	Free & Open Source (3)	Add-on Functionality (3)	Shortlisted
1	KwikiKwiki	X	X	-	-	-	-	-	X
2	DokuWiki	X	X	-	-	-	-	-	X
3	JSPWiki	X	X	X	-	-	-	-	X
4	PerSpective	X	X	X	-	-	-	-	X
5	FlexWiki	X	X	X	-	-	-	-	X
6	UseModWiki	X	X	X	X	X	-	-	X
7	OddMuseWiki	X	X	X	X	X	-	-	X
8	TeleparkWiki	X	X	X	X	X	X	-	X
9	MoinMoin	X	X	X	X	X	X	X	X
10	PmWiki	X	X	X	X	X	X	X	X
11	TWiki	X	X	X	X	X	X	X	X
Information Source		(1) http://c2.com/cgi-bin/wiki?WikiEngines (2) http://c2.com/cgi-bin/wiki?TopTenWikiEngines (3) http://www.wikimatrix.org							

The following additional criteria were used to select the final wiki: concurrent editing features, executable code size, process memory footprint, directory structure, and functionality design. Table 2 summarizes the outcome of the selection analysis. TWiki is deemed to be the more suitable for implementation in the MYSEA test-bed environment. The following paragraphs outline these criteria in greater detail.

Table 2: Summary of selection process

S/N	Description	PmWiki	TWiki
1	Concurrent editing	Supported.	Supported. More robust implementation.
2	Size of executable code	841 KB	2250 KB
3	Footprint of process memory	4 MB	6 to 13 MB
4	Directory structure	Flat. Supports the organization of wiki pages by classification only.	Hierarchical. Supports both the organization of wiki pages by topic and classification.
5	Functionalities design	Accountability not enforced in access control.	Accountability is strictly enforced in access control.

* Shaded box denotes the preferred choice.

PmWiki users are unaware of concurrent editing until an attempt is made to save changes, at which point conflicts are highlighted and the user may resolve conflicts prior to permanently saving the text. TWiki warns users at the start of an attempt to concurrently edit a document. It also prompts users for decisions when conflicts within the document are found. The total size of the PmWiki main PHP script file and 33 supporting scripts is 841Kbytes. For TWiki, the total size of its 230 script files is 2262Kbytes. If size is as an indicator relative code complexity, PmWiki is smaller and perhaps less complex. The *size* variable given by Linux *top* command was used to determine the memory consumption and TWiki was found to require more memory. PmWiki supports only one level of sub-directories and uses a structured file naming convention to embed further hierarchical information within the filenames, as illustrated in Figure 2, where # indicates the start of a comment.

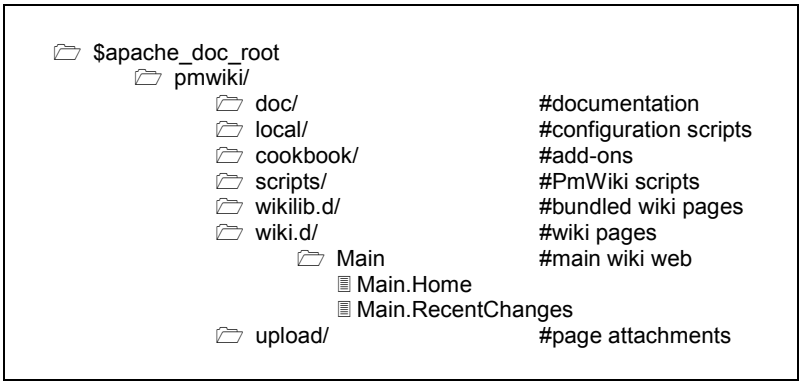

Figure 2: PmWiki directory structure

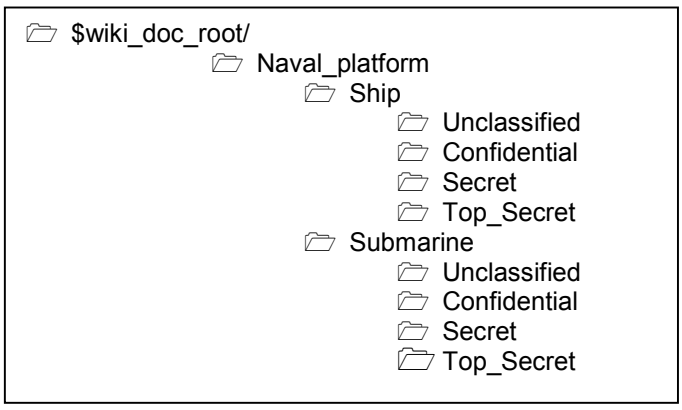

Figure 3: TWiki directory structure

For TWiki, its logical hierarchical grouping is realized in a corresponding hierarchical directory structure, as shown in Figure 3.

Mandatory policy may be accommodated in the wiki file and directory organization in one of two ways. As shown in Figure 4, the first organizes the wiki pages by topic and then by classification.

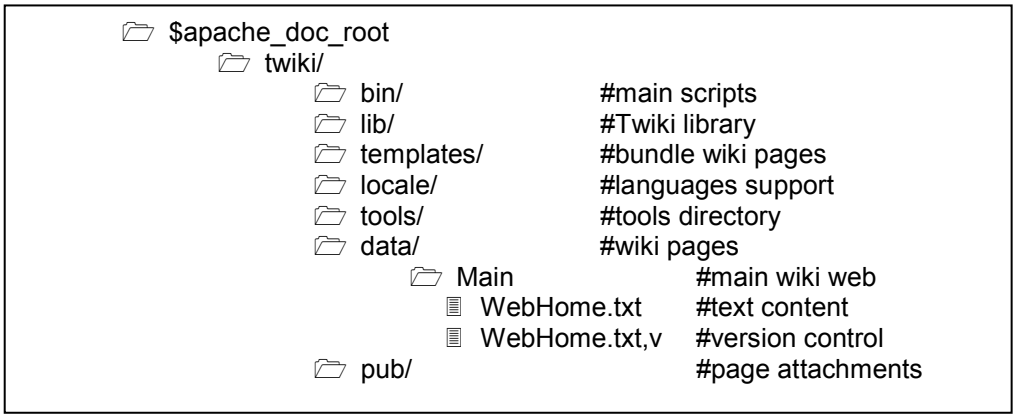

Figure 4: Organization of Wiki Pages by Topic.

The second organizes the wiki pages by classification and then by topic, as illustrated in Figure 5.

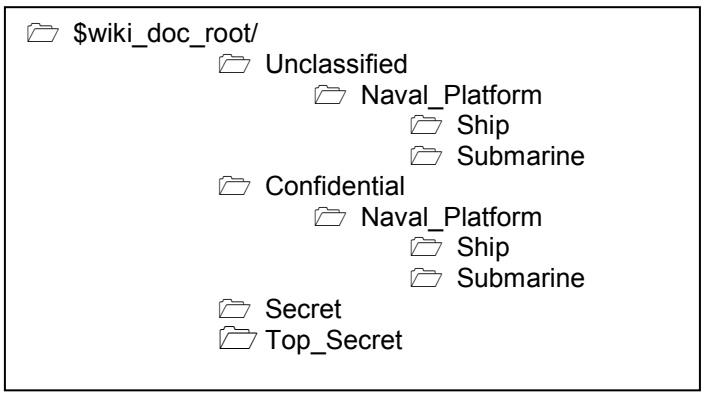

Figure 5: Organization of Wiki Pages by Classification.

The hierarchical directory structure of TWiki supports both approaches. PmWiki can only support the organization of wiki pages by classification. Furthermore, all PmWiki pages within each classification sub-directory must be stored in a flat file structure, differentiated only via the filename convention, which would result in long file names. Long file names are inadvisable because STOP OS filenames are limited to 256 characters. In addition, extremely long filenames complicate file administration.

Although the limitations of PmWiki are not insurmountable, the more intuitive organization possible with TWiki better suited for our anticipated operational environment.

PmWiki allows multiple logins within the same browser session. Login names are not displayed anywhere within the wiki page to indicate the current active login. This could create confusion regarding the session status.

The most notable difference between the two wiki engines is their history logging mechanisms. PmWiki allows users to arbitrarily specify the author name during the editing session. Hence, accountability is not enforced through the PmWiki's implementation. TWiki enforces one login session per browser. Login sessions are distinguished through the display of the login ID on every page request, and the history log entries are bound to the login ID. In this respect, TWiki is considered to be superior.

3. Concept of operation

In the MYSEA Test-bed environment, users need to be authenticated via the Trusted Path Extension (TPE) before connection to any services in the MLS server is permitted. The TPE is a special purpose high assurance component juxtaposed between the untrusted client and the high assurance MLS server. Following identification and authentication, users negotiate a session level from the range of security levels for which they are authorized. A user's maximum session level will be bounded by his or her clearance. The session level determines the level of access to data that are provided by the services in the MYSEA Test-bed, including the wiki, through the enforcement of MAC policy by the STOP 6.3 operating system.

Using any web browser, the user can log into the wiki. At this point, the wiki server enforces the wiki-specific DAC policy for the wiki session. The user can read, edit or create wiki pages as constrained by both the DAC policy enforced by the wiki server, and the MAC policy enforced by the underlying operating system.

The following three example scenarios illustrate the possible collaboration between two users in a multilevel environment.

Example Scenario 1: For users, Sally and Jim, who have the same mandatory session level and are simultaneously editing the same page, the wiki merges the saved results and the last person to save will be presented with the conflicting sections. He or she will have to decide on how to resolve the conflict.

Example Scenario 2: Alice is logged in at the SIM_CONFIDENTIAL session level, while Bob is logged in with the SIM_UNCLASSIFIED session level. As Alice is the secretary of a coalition meeting, she is uploading the meeting presentation materials, which are at the SIM_CONFIDENTIAL level. Alice is able to view, but not modify, information on the wiki labeled at the SIM_UNCLASSIFIED level. Bob is an invited attendee of the meeting, and he can browse the meeting agenda page at the SIM_UNCLASSIFIED level. However, Bob is unable to see the meeting presentation materials uploaded by Alice, which have a security level dominating his session level; in fact, he will not even know that the upload area exists.

Example Scenario 3: During the board meeting, Bob was asked to provide some documents to the committee. He logs in at the SIM_UNCLASSIFIED session level, and uploads the documents to the designated folder. Alice and the other meeting participants who are logged in at the SIM_CONFIDENTIAL session level will be able to read the documents uploaded by Bob.

4. Porting methodology

The implementation of the wiki engine on MYSEA test-bed was a port of the wiki engine and was conducted in stages. It began with the installation of the wiki engine on RedHat 8.0 Linux on an Intel-based machine, followed by its installation on the XTS-400 as a single-level process confined to a single-level domain, and finally the wiki was implemented as a multilevel-aware application on the XTS-400 server that could use the multilevel capabilities of the underlying operating system. During each stage, testing was performed.

For the multilevel implementation, the Apache-like web server is set up as an inetd-like service, bound to a specific port and initiated by the MYSEA Secure Session Server (SSS) daemons. The SSS processes spawn new processes at the security and integrity level of the incoming request. Two deflection directories for storage of temporary and deleted files were created. Various wiki web directories at different security levels were also created.

5. Wiki design and architecture

A typical TWiki website is organized into *topics*, each of which constitutes a wiki page, and *webs*, which are a collections of related topics. Users create new topics, and edit existing topics as allowed by DAC policy (Wikipedia 2007b). As shown in Figure 6, a web is a sub-directory within the main *data* directory, and a topic is a file within a web sub-directory. For each topic, there is a

corresponding version control file. Users can upload and download files to and from the wiki pub directory, the organization of which parallels that of the *data* directory.

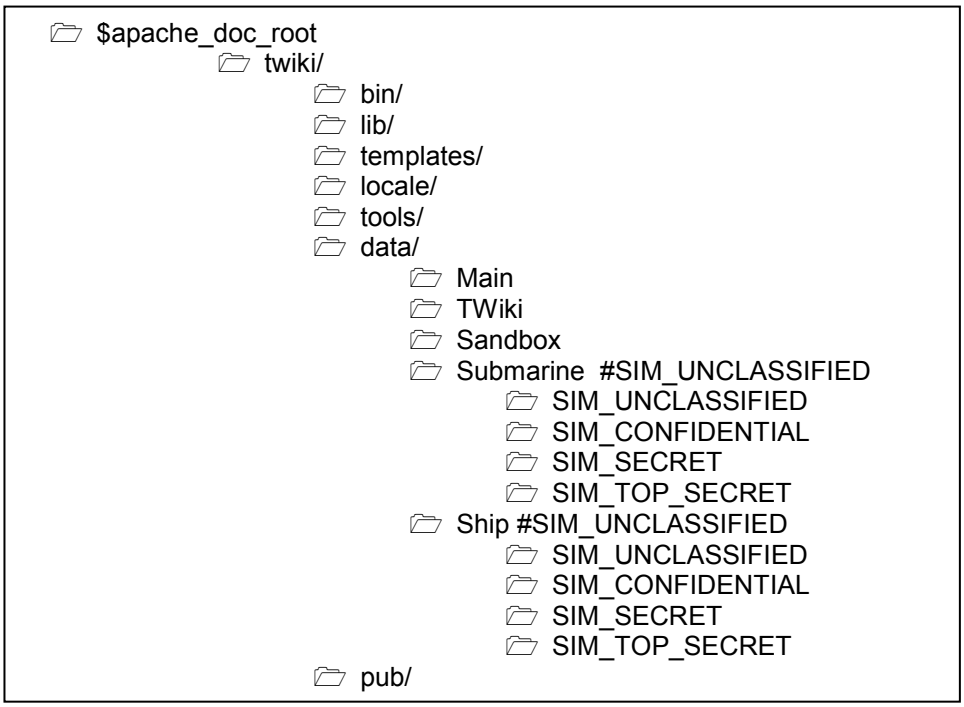

Figure 6: TWiki directory structure

Within the various webs, sub-webs can be created in a hierarchical manner. TWiki also requires a */tmp/twiki* directory for temporary files, and a *…/$apache_doc_root/twii/data/Trash* directory for deleted files.

5.1 MLS Wiki architecture

In the MLS environment, the */tmp/twiki* directory is created in each sensitivity level-specific deflection directory of */tmp*. This ensures that users logged in at different session levels are able to write to */tmp/twiki* without information leakage. *Trash* is also created as a deflection directory for the same reason.

TWiki is able to support the organization of wiki content by classification or by topic. For organization by classification, webs corresponding to the various security classification levels are created under the *data* and *pub* directories at the desired sensitivity levels. Sub-webs corresponding to different topics are then created within each of these webs. In contrast, for organization by topic, webs corresponding to each topic of interest are created under the *data* and *pub* directories at the lowest sensitivity level (i.e., SIM_UNCLASSIFIED). Sub-webs are then created within each of these content webs at the desired sensitivity levels. In this approach, all the possible login session levels must be anticipated and the corresponding web directories created in advance. Sub-webs within these web directories can be created when required by the administrator.

Figures 7 and 8 illustrate the directory structure for organization by classification and organization by topic respectively.

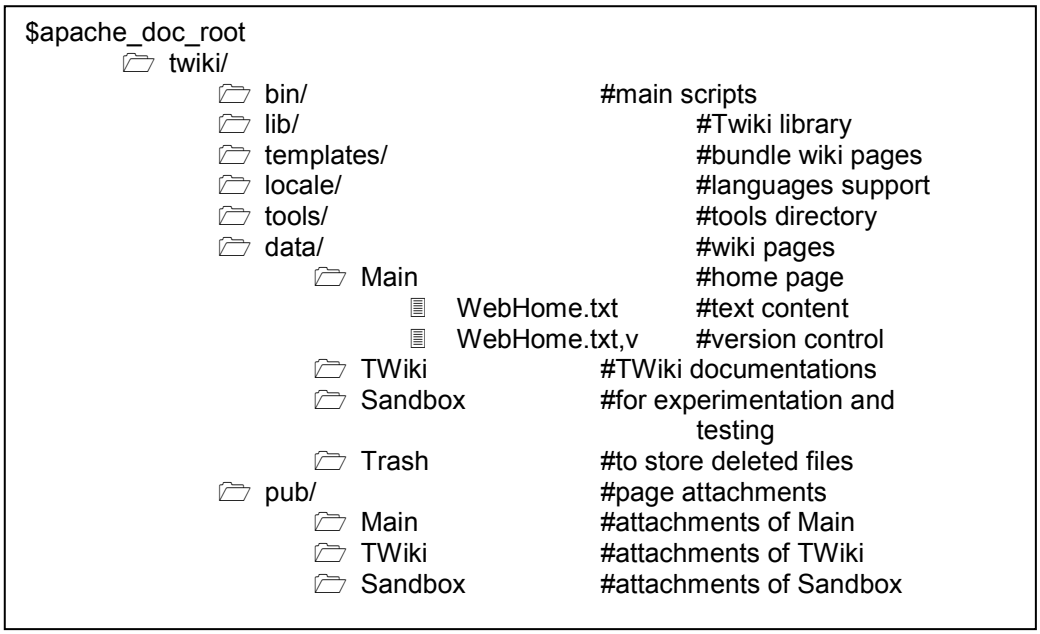

Figure 7: TWiki data directory organization by topic

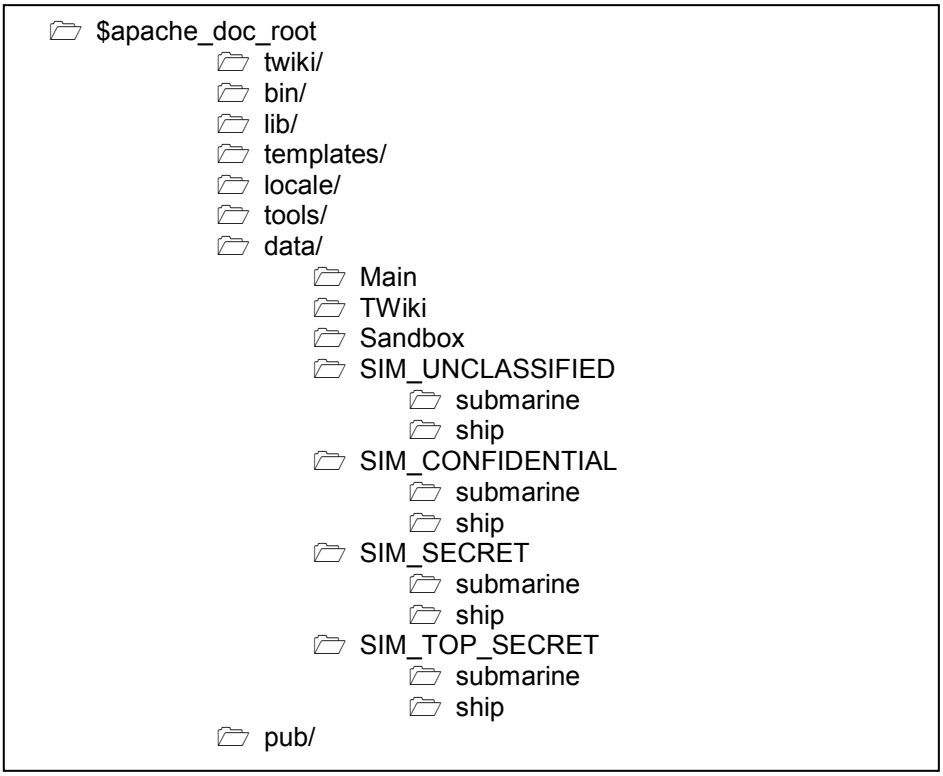

Figure 8: Twiki data directory organization by classification.

In the MLS wiki, a user can create a link within a wiki page at the current session level to a wiki page with a sensitivity level higher than the current session level. TWiki will show a question mark "?" beside the newly created link to indicate that the target page does not exist. This is standard TWiki behavior when links are created to non-existent wiki pages. If the user attempts to access the newly created link, the TWiki engine informs the user that the higher security level *web* does not exist and provides an option to create it. Attempts to create the *web* at the higher level will be denied, as write-up is not permitted.

5.2 Wiki porting Cchallenges in MYSEA

TWiki enforces an application-level access control mechanism that allows users to set permissions on files they have created. In the standard TWiki configuration, the Apache server is expected to run as a special user, typically named *apache*, and all wiki files are readable and writeable only by *apache*. In MYSEA, a daemon instantiates server processes on behalf of each authenticated user. Thus the user ID associated with the server process is that of the logged in user and any files created by the server process are owned by that user. This differs from the standard Linux environment, where the server processes commonly assume a dedicated user ID, and any files created by those processes are owned by that user ID. Thus, in MYSEA, all users would have to belong to the same group and have read/write permissions set for that group so that each web server instance could access the wiki files. TWiki would then enforce its usual application-level access control mechanism.

To implement the required changes, an attempt was made to change the default permission of new text files to rw-rw-r-- by setting the *umask* to 002 in the */etc/profile script*. This works for the interactive console logon session, but not for the Apache daemon process, as it does not inherit the *umask* setting from the */etc/profile script*. It was therefore decided to modify the TWiki code to include a *umask* command that changed the default permission on new files to rw-rw-r--.

As is often the case with ad hoc repairs, this introduced a new vulnerability: users could login through an interactive shell session and by-pass the TWiki access controls to gain access to the wiki files. Two approaches to overcome this limitation were considered. The first restricts the users' ability to access files through interactive shell sessions, whereas the second uses the XTS access control lists to restrict file access.

The ability to launch interactive shell sessions can be restricted by ensuring that services such as SSH and telnet are not enabled. In MYSEA, SSH and telnet services are not supported, but users are allowed to launch interactive shell sessions via the WebShell CGI program. Although WebShell does not allow execution of interactive programs such as the *vi* editor, users can still modify the contents of a file by techniques such as output redirection. Also, the MYSEA WebDAV implementation allows users to navigate to their home directories, and consequently to the Apache document root located under /home/http/htdocs. Hence users would be able to view and edit files under the TWiki data directory using WebDAV.

Since WebShell and WebDAV are part of the overall concept of operations of MYSEA, their removal is not a viable option. However, a separate server intended exclusively for wiki services provides a solution. A single-sign-on framework for MYSEA has been proposed (Bui 2005) and will enhance usability in this federated-server approach.

5.3 XTS access control list

In the STOP 6.3 operating system, up to seven entries can be added to each file and directory ACL. Each can be a combination of user names and group names.

A subset of the TWiki access modes can be mirrored using XTS ACL features. TWiki supports three access modes to topics and webs: view, change and rename. The view access mode corresponds to the read access mode of the STOP OS, whereas the change and rename access modes correspond to the write access mode. Clearly, DAC granularity is lost if the XTS ACL mechanism is adopted to implement the default TWiki DAC mechanism. Additionally, the revision control file, which is used to generate the history log, would require a parallel ACL treatment. Thus, any authorized users of the wiki topic would be able to modify the history log. Finally, the TWiki password file would be vulnerable to user modification, although this could be mitigated by ensuring that the administrator is the owner of the password file, thus prohibiting users from modifying it.

If implemented at the directory level, all files within the directory have owner and group read/write permissions enabled; each user will either have read and write permission to all files or to none.

At the file level, authorized users are added to the ACLs of each topic file and to its corresponding version control file. As users have ability to create their own topics, the number of topic files and the associated version control files can be large. Management of these ACLs would not scale well, overburdening the system administrator. Therefore, new code for insertion and deletion of ACL entries as well as modification of the standard TWiki scripts to interface with the newly developed codes would be necessary.

Neither of the ACL-based solutions is recommended. Instead, we opted to host the wiki on a separate, dedicated server where interactive shell sessions are not possible. This solution preserves all wiki functionality and wiki engine upgrades or changes are possible with minimal modification.

6. Implementation

To implement the federated design supporting TWiki in the MYSEA environment, two MYSEA servers were used: one as the standard MYSEA MLS server and the other as the TWiki server.

The directory structure of the wiki server was organized by classification, as illustrated in Figure 9. The *pub* directory was configured with a similar directory structure.

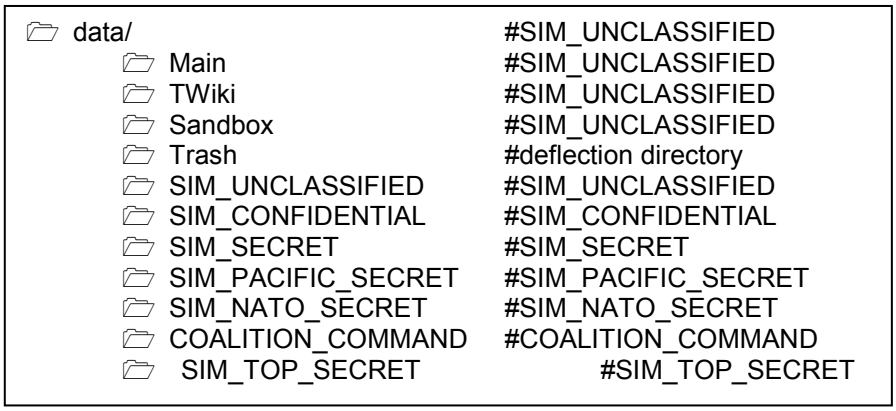

Figure 9: Directory structure

The *data* and *pub* directories, and all the sub-directories within them, were created with file permissions as follows: owner:rwx; group: rwx; public:r-x. All files under *data* and *pub* have file permissions set to owner:rw-; group: rw-; public:r--.

So that new webs and topics created by the wiki engine have the permission settings described above, the source code of the wiki engine was modified. In particular, a “umask=002” command was inserted into the manage.pm and save.pm Perl modules.

To configure the server, the WebDAV module was removed via recompilation of Apache. The WebShell GGI script was also removed so that users are unable to execute interactive shell sessions on the wiki server.

7. Testing

Testing consisted of four parts: functional tests, DAC security tests, MAC security tests and integration tests. Functional testing confirmed that the wiki provided its expected services and were conducted for all three configuration categories: Linux, single level XTS and multilevel XTS. DAC security tests were performed for all configurations to verify that the DAC policies are being enforced correctly. MAC security tests were also performed for the single level XTS and multilevel XTS configurations to verify that the MAC policies are being enforced correctly.

Integration testing verified that the wiki server was able to function correctly in the MYSEA environment, and that users were able to use MYSEA services available on all servers. Currently users must login to the servers independently.

8. Summary, related and future work

We have designed a MLS-aware high assurance wiki architecture, and have demonstrated its implementation in the MYSEA test-bed using TWiki. Testing confirmed that the MLS-aware wiki server is fully functional and is properly constrained by the underlying MAC and DAC policies enforced by the STOP OS. The availability of this highly popular collaborative tool in a multilevel environment will offer an improved method for sharing information in an environment constrained by mandatory enforcement of national security policies.

8.1 Related work

The Trusted Services Engine (TSE) implements WebDAV on a MILS separation kernel (McNamee et al. 2006). Although details are not publicly available, but it is understood that Galois plans to construct a wiki with features such as read down and cross-domain fusion. The MYSEA wiki server supports read down capability, and a cross domain fusion capability is planned.

8.2 Future work

TWiki maintains its own database of user IDs and passwords. Currently, there is no synchronization between the XTS system's IDs and passwords, and those of TWiki. A single sign-on solution would provide a more usable interface.

Cross domain content merging would enable users to see relevant *topic* contents up to their established session level, instead of having to browse to webs at different security levels to view a topic at different sensitivity levels. Such a fusion capability would provide a better unified view of related content, thus enhancing the user experience and productivity.

To avoid the confusion caused by multiple sets of user mail boxes on the MYSEA servers, the wiki server should be configured SMTP and IMAP services.

Acknowledgements

This work was sponsored by the SPONSOR-GIVEN-LATER. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring organization.

References

Aronsson, L. (2007) "Operation of a Large Scale, General Purpose Wiki Website", in Proceedings of the 6th International ICCC/IFIP Conference on Electronic Publishing, November, pp 27-37.

Bell, D.E. and LaPadula, L.J. (1976) "Secure Computer System: Unified Exposition and Multics Interpretation", ESD-TR-75-306, Electronic Systems Division (AFSC).

Biba, K.J. (1977) "Integrity Considerations for Secure Computer Systems", MTR-3153, Mitre Corp., Bedford, M.A, April.

Bradney, J. A. (2006) "Use of WebDAV to Support a Virtual File System in a Coalition Environment", Master's Thesis, Naval Postgraduate School, Monterey, CA, June.

Bui, S. (2005) "Single Sign-on Solution for MYSEA Services", Master's Thesis, Naval Postgraduate School, Monterey, CA, September.

Carstensen, P. H. and Schmidt, K. (1999) "Computer Supported Cooperative Work: New Challenges to Systems Design", In K. Itoh (Ed.), Handbook of Human Factors.

CISR (2007) MYSEA poster, http://cisr.nps.navy.mil/downloads/07poster_MYSEA.pdf, last viewed 1 November 2007.

Cunningham (2007) "Wiki Engines," [online], Cunningham and Cunningham, Inc., http://c2.com/cgi-bin/wiki?WikiEngines, last viewed 2 March 2007.

Getronics Government Solutions (2002) XTS-400, STOP 6.0, User's Manual, Document ID: XTDOC0005-01, Getronics Government Solutions, LLC, Herndon, VA, August.

Irvine, C. E., Levin, T. E., Nguyen, T. D., Shifflett, D. J., Khosalim, J., Clark, P. C., Wong, A., Afinidad, F., Bibighaus, D., and Sears, J. (2004) "Overview of a High Assurance Architecture for Distributed Multilevel Security", in Proceedings of the 2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop, West Point, NY, June, pp 38-45.

B. W. Lampson (1973) "A note on the confinement problem," Communications of the A.C.M., Vol 16, No.10, pp. 613–615.

McNamee, D., Heller, S. and Huff, D. (2006) “Building Multilevel Secure Web Services-Based Components for the Global Information Grid”, STSC CrossTalk, May.

Nguyen, T.D., Levin, T. E. and Irvine, C. E. (2005) “MYSEA Testbed,” in Proceedings of the 6th IEEE Systems, Man and Cybernetics Information Assurance Workshop, West Point, NY, June, pp. 438-439.

PmWiki (2007) “PmWiki” http://www.pmwiki.org/wiki/PmWiki/PmWiki, [online], dated 26 March 2006, last viewed 08 November 2007.

TWiki (2007a) “TWiki Access Control,” http://twiki.org/cgi-bin/view/TWiki/TWikiAccessControl, [online], dated 27 September 2007, last viewed 30 October 2007.

TWiki (2007b) “TWiki,” [online], http://www.twiki.org/, last viewed 08 November 2007.

Wikimatrix (2007) http://www.wikimatrix.org/, [online], last viewed 2 March 2007.

Wikipedia (2007a) “Wiki,” http://en.wikipedia.org/wiki/Wiki, [online], dated 30 July 2007, last viewed 30 July 2007.

Wikipedia (2007b) “File System Permission,” “http://en.wikipedia.org/wiki/Unix_permission”, [online], dated 22 November 2007, last viewed 26 November 2007.

Wikipedia (2007c) “Comparison of wiki software,” [online], http://en.wikipedia.org/wiki/Comparison_of_wiki_software, dated 1 March 2007, last viewed 2 March 2007.